Parallel Mesh Generation With a Global Change in the Marching Cubes Algorithm

Aruquia Peixoto* [1], Thiago Franco Leal[2] and Carlos A. de Moura[3]

[1] CEFET/RJ, Department of Basic Disciplines, aruquia@gmail.com
[2] UERJ, Rio de Janeiro State University, Brazil, thiago_tfl@hotmail.com
[3] UERJ, Rio de Janeiro State University, Brazil, demoura@ime.uerj.br

Key Words: *Mesh Generation, Implicit Surface, Parallel Programming in GPU.*

In this work we propose a parallel method implemented in GPU environments to generate meshes from Implicit Surfaces that turn out to be more appropriated for engineering applications, like the ones based on Finite Element Methods. This work improves the article presented at WCCM 2012 which made a general optimization on Marching Cubes Algorithm, generating triangles with better quality through better relation between their edges and their angles.

Marching Cubes is a well known algorithm used to generate meshes formed by triangles. Despite its simplicity to be implemented and understood, this algorithm can generate triangles with very small angles, and as a consequence, with one edge much smaller than the others, as shown in Figure 1, a). We propose a parallel method to improve the mesh in two steps, each being done in a GPU environment.

In the first step the entire cube is divided in smaller cubes, the voxels, and they are polygonized in parallel. For each voxel, if the surface intersects it, then a modified Marching Cubes table is used to generate only the border of the surface portion that intersects this voxel. After generating the border, if it has a corner with a small angle, its neighbours are connected to prevent this vertex to be divided, and this voxel is marked. This small angle will be processed at the second step.

The remaining polygon is evaluated in order to insert a vertex inside this polygon. If a new vertex is inserted, all vertices on the polygon border will be connected to this new vertex, thus generating the triangles. Otherwise, the mesh is generated by connecting the vertex with the biggest angle to one of its two neighbours, that are at a two edge distance, the one with the biggest angle being chosen. This process is repeated until all mesh triangles inside this voxel are generated.

The second step consists in repositioning all vertices that have a triangle with a too small angle and that was marked at the first step. This step is also run in parallel and the new position, due to all the vertices that are directly connected to this vertex with a smooth correction, as implied by the implicit surface, is used. The result of the global change can be seen in Figure 1 b): the triangles generated by the proposed method have better angles and better ratio between their sides.

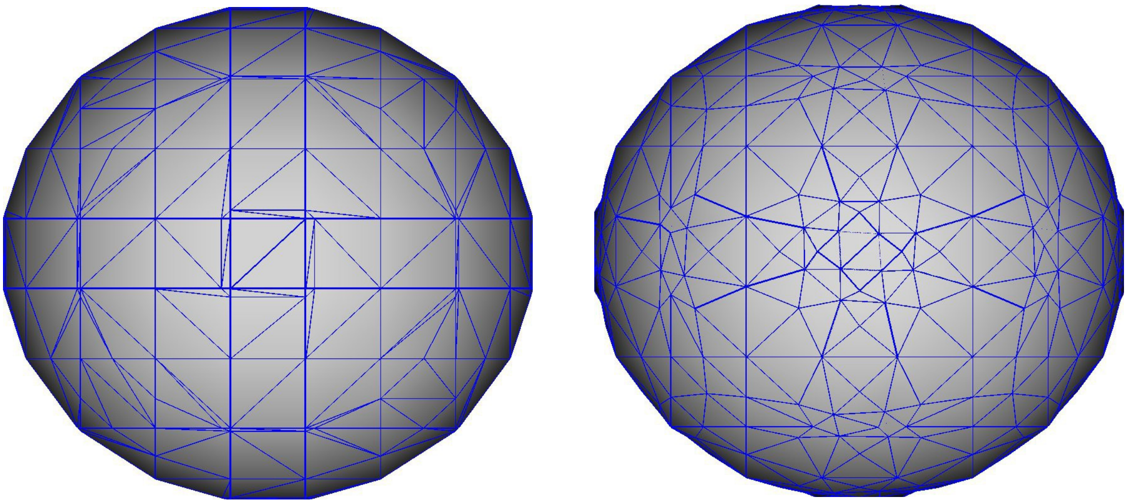

Figure 1. a) Sphere generated by the Marching Cubes algorithm. b) Sphere generated by the algorithm proposed in this work.

REFERENCES

[1] T. Leal, A. Peixoto, M. Dreux and C. de Moura. Generation of triangular meshes through vertices connection modification of the marching cubes algorithm. In: *11th USNCCM. Minneapolis: US Association for Computational Mechanics*, 2011. p. 120.

[2] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface reconstruction algorithm. In: Proceedings of SIGGRAPH. Anaheim: ACM SIGGRAPH Computer Grphics, 1987. p. 163-169.

[3] A. Peixoto, T. Leal, M. Dreux and C. de Moura. A global change for mesh improvements in the marching cubes algorithm. In: *10th WCCM*, 2012, Sao Paulo. p. 210

[4] J. R. Shewchuk, What Is a Good Linear Finite Element? - Interpolation, Conditioning, Anisotropy, and Quality Measures, In *Proc. of the 11th International Meshing Roundtable*, 2002